**设计模式**

前面讲过，社会化的分工越来越细，自然在软件设计方面也是如此，因此对象的创建和对象的使用分开也就成为了必然趋势。因为对象的创建会消耗掉系统的很多资源，所以单独对对象的创建进行研究，从而能够高效地创建对象就是创建型模式要探讨的问题。这里有6个具体的创建型模式可供研究，它们分别是：

简单工厂模式（Simple Factory）；

工厂方法模式（Factory Method）；

抽象工厂模式（Abstract Factory）；

创建者模式（Builder）；

原型模式（Prototype）；

单例模式（Singleton）。

说明：严格来说，简单工厂模式不是GoF总结出来的23种设计模式之一。

**2．结构型模式**

在解决了对象的创建问题之后，对象的组成以及对象之间的依赖关系就成了开发人员关注的焦点，因为如何设计对象的结构、继承和依赖关系会影响到后续程序的维护性、代码的健壮性、耦合性等。对象结构的设计很容易体现出设计人员水平的高低，这里有7个具体的结构型模式可供研究，它们分别是：

外观模式（Facade）；

适配器模式（Adapter）；

代理模式（Proxy）；

装饰模式（Decorator）；

桥模式（Bridge）；

组合模式（Composite）；

享元模式（Flyweight）：

1、意图：通过共享有效支持大量细粒度的对象，来提供应用程序的性能，节省系统中重复创建对外实例的性能消耗。

**3．行为型模式**

在对象的结构和对象的创建问题都解决了之后，就剩下对象的行为问题了，如果对象的行为设计的好，那么对象的行为就会更清晰，它们之间的协作效率就会提高，这里有11个具体的行为型模式可供研究，它们分别是：

模板方法模式（Template Method）；

观察者模式（Observer）；

状态模式（State）；

策略模式（Strategy）；

职责链模式（Chain of Responsibility）；

命令模式（Command）；

访问者模式（Visitor）；

调停者模式（Mediator）；

备忘录模式（Memento）；

迭代器模式（Iterator）；

  解释器模式（Interpreter）。